**Naval Research Laboratory**
**Washington, DC 20375-5320**

**NRL/MR/5524--17-9751**

# High Performance Computing Modernization Program Kerberos Throughput Test Report

Daniel G. Gdula
*KEYW Corporation*
*Hanover, Maryland*

Joseph A. Molnar
*Networks and Communication Systems Branch*
*Information Technology Division*

October 26, 2017

Approved for public release; distribution is unlimited.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

**1. REPORT DATE** *(DD-MM-YYYY)*
26-10-2017

**2. REPORT TYPE**
Memorandum Report

**3. DATES COVERED** *(From - To)*
01 October 2016 – 01 August 2017

**4. TITLE AND SUBTITLE**

High Performance Computing Modernization Program
Kerberos Throughput Test Report

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**
633461

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**

Daniel G. Gdula* and Joseph A. Molnar

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**
55-0124

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Naval Research Laboratory, Code 5524
4555 Overlook Avenue, SW
Washington, DC 20375-5320

**8. PERFORMING ORGANIZATION REPORT NUMBER**

NRL/MR/5524--17-9751

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
DOD HPC Modernization Program
CEERD-IZP
USAERDC, Waterways
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

**10. SPONSOR / MONITOR'S ACRONYM(S)**

HPCMP

**11. SPONSOR / MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION / AVAILABILITY STATEMENT**

Approved for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**
*KEYW Corporation, Hanover, MD

**14. ABSTRACT**

Kerberos is a network authentication protocol developed at MIT to provide encryption for client/server functions. The team at the HPCMP maintains an internally developed, adapted version of an older MIT Kerberos release for user access via the network to the HPCMP supercomputers. The HPCMP has re-baselined the HPCMP Kerberos to the most current MIT Kerberos release, and has incorporated HPCMP unique functionality as Kerberos plugins. These plugins include Yubikey authentication, PIPE credential caching, and support for DoD X.509 certificates. Throughput testing was required to determine if file transfer speeds improved with the pre-release production kit. Throughput performance was measured on a variety of operating systems employed by HPC users.

**15. SUBJECT TERMS**

| | | |
|---|---|---|
| Kerberos | SSH | Supercomputer |
| Authentication | FTP | |
| File transfer performance | FileZilla | |

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| | | | | | Joseph A. Molnar |
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** | | | **19b. TELEPHONE NUMBER** *(include area code)* |
| Unclassified Unlimited | Unclassified Unlimited | Unclassified Unlimited | Unclassified Unlimited | 26 | (202) 767-0327 |

**Standard Form 298 (Rev. 8-98)**
**Prescribed by ANSI Std. Z39.18**

## CONTENTS

## FIGURES

## TABLES

# HIGH PERFORMANCE COMPUTING MODERNIZATION PROGRAM KERBEROS THROUGHPUT TEST REPORT

## 1. INTRODUCTION

### 1.1 Kerberos

Kerberos is a network authentication protocol developed at MIT to provide encryption for client/server functions [1]. The team at the HPCMP maintains an internally-developed, adapted version of an older MIT Kerberos release for user access via the network to the HPCMP supercomputers. The HPCMP has re-baselined the HPCMP Kerberos to the most current MIT Kerberos release, and has incorporated HPCMP unique functionality as Kerberos plugins. The pre-release production kit was used in these tests to compare against the current release kit.

YubiKey support was added to incorporate a different method for login authentication. YubiKey is a small USB device that supports various encryption protocols that support two-factor authentication. Users can employ a YubiKey as an alternative to the standard method that requires a common access card (CAC) to log in to an HPCMP system. This strategy provides for an easy, cost-efficient way to allow users access for individuals who do not qualify for a CAC, or require an alternative method due to requirements of their work.

Finally, HPCMP Kerberos incorporates modified pkinit support. Standard configuration relies on Kerberos-specific extensions in certificates. This modified support module allows users to use their DoD X.509 certificates. It also allows for more flexible Kerberos configuration files. This results in a more robust method. The identified modules were developed by the HPCMP developers and are added modules to the MITs Kerberos client package, which, when integrated, form the HPCMP pre-release production client.

### 1.2 High Performance Computing Modernization Program

The High Performance Computing Modernization Program (HPCMP) is an initiative to provide the Department of Defense (DoD) the means to research and process data using high-performance computing sites around the country (see Figure 1). These sites allow scientists and researchers from institutions such as colleges or military research installations to schedule data processing jobs and run complex simulations to produce data related to their projects. These HPCMP sites provide computing solutions around the country, which are connected via the Defense Research and Engineering Network (DREN) wide-area network. This backbone network is how researchers transfer high volumes of data to and from an HPCMP supercomputer. [2]

Each HPCMP supercomputer site is accompanied by a utility server. These servers are Linux-based, and used as software development platforms for remote management of batch jobs, as file share utilities among HPCMP systems, and other many other things. These utility servers also provide a maintenance platform for administrators. While these servers offer a useful stepping stone, they do not supplement the high computing power of the main supercomputer. Each supercomputer is different in node architecture as well as hardware specifications.

Manuscript approved August 9, 2017.

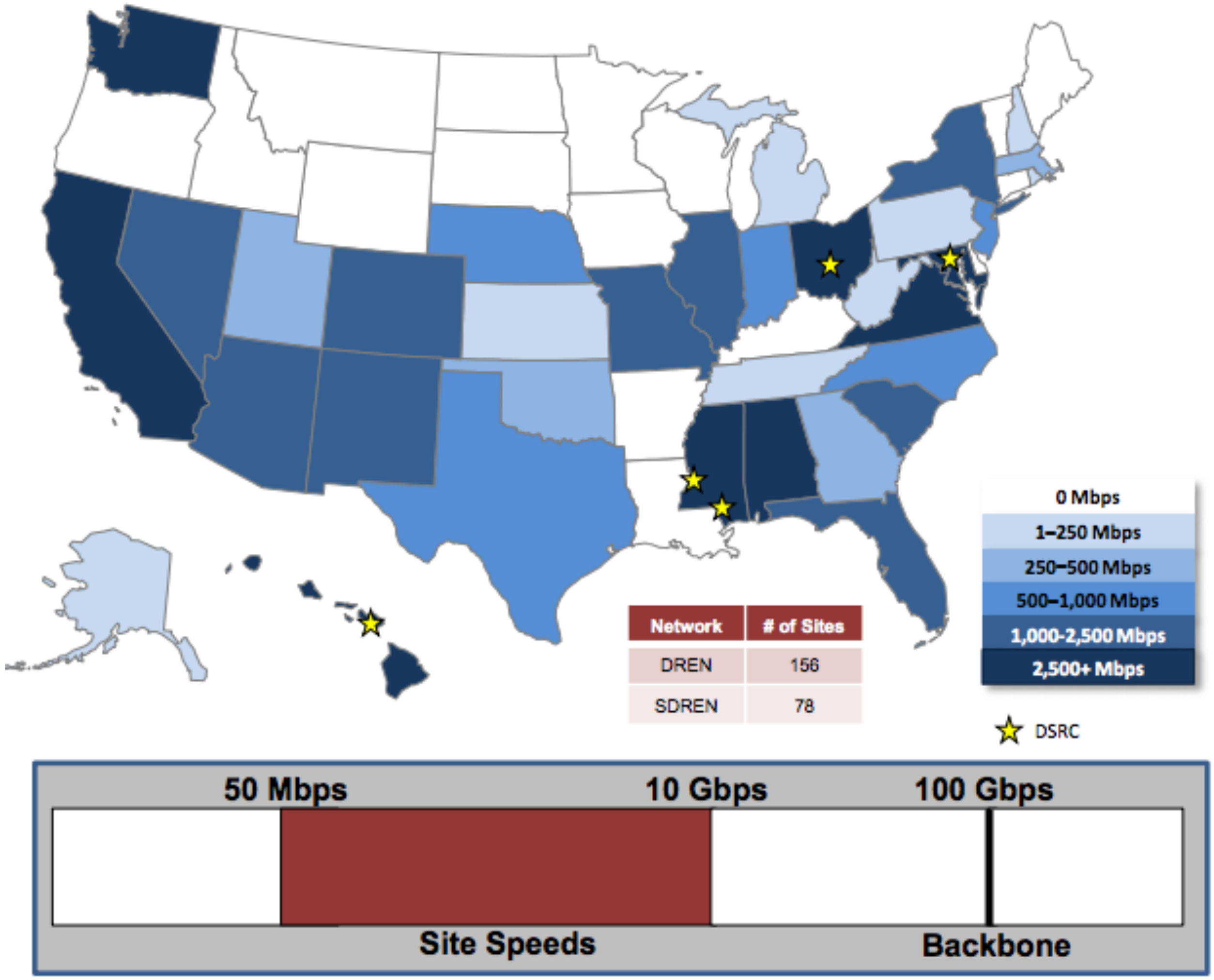



Fig. 1: HPCMP computing sites

Table 1: HPCMP supercomputer specifications

| ARL Excalibur | MHPCC Riptide | NAVO Haise | AFRL Spirit | ERDC Garnet | NRL Polar |
|---|---|---|---|---|---|
| CrayXC40<br>99,136 cores<br>Cray Linux<br>9.92ms roundtrip | IBM iDataPlex<br>12,096 cores<br>Redhat Enterprise<br>158.42ms roundtrip | IBM iDataPlex<br>19,520 cores<br>Redhat Enterprise<br>pings disabled | SGI ICE X<br>73,440 cores<br>Redhat Enterprise<br>pings disabled | Cray XE6<br>150,912 cores<br>Cray Linux<br>40.43ms roundtrip | SGI Altix Ice<br>2048 cores<br>Redhat Enterprise<br>0.87ms roundtrip |

There is more than one supercomputer at each HPCMP location, which may have different specifications, or is newer in implementation. Each HPCMP location is connected via the DREN wide-area network, which is comprised of a 100Gbps backbone. The sites themselves, however, may only have an interface of 10-40Gbps to the backbone network. This varies from site to site. Each HPCMP supercomputer will also vary on user utilization on any particular day. However, this only impacts jobs currently being executed or scheduled, not network connectivity.

# 2. THROUGHPUT TESTING

## 2.1 Testing Components

Throughput testing was done to determine the benefits of the pre-release production kit compared to the current release kit. These kits are comprised of a few transfer tools to get files to and from the HPCMP supercomputers. These tools are the FileZilla client as well as a PuTTY command line interface tool. Each tool is able to use the two main transfer protocols that were tested, File Transfer Protocol (FTP) and Secure File Transfer Protocol (SFTP).

FileZilla is a cross-platform, file transfer client. It is used as a configurable client to access files and manage data on an FTP, server provided you have the credentials to access the system. This is an open-source, C++ application released in June 2001 and hosted on the Internet repository "SourceForge" [3].

PuTTY is a network login application capable of SFTP, SCP, SSH, and other protocols. It allows tunneling, port-forwarding, and proxy configuration. It emulates a console environment and consists of a few separate modules such as PSCP, PSFTP, Plink, and the main module. These modules combined make up the complete application that offers a multitude of tools. It is also open-sourced and was released in November of 1998 [4].

When using either of these applications, the connection between the host and remote site is not encrypted in most cases. In this case, we use a Kerberized File Transfer Protocol (KFTP) in order to protect the connection [5]. This protocol has the base of normal FTP, but refrains from sending Kerberos login information over ther network that could potentially be sniffed. Using KFTP, authentication is achieved with a Kerberos ticket to initiate a connection to an offsite computer, in this case an HPCMP supercomputer. This ticket can be obtained using a few methods such as a CAC or, as an alternative, YubiKey.

## 2.2 Testing Parameters and Procedure

Tests were executed with data volume in mind. Two different file sizes were utilized, 50MB and 3GB. A normal data transfer to one of these supercomputers can range from multiple gigabytes to terabytes. The tests were run from various operating systems, but the main focus was on Microsoft Windows 7, Microsoft Windows 8.1, and Microsoft Windows 10. In order to have a comparison on the operating system level, both Mac (El Capitan) and Ubuntu (16.04) were also used as a contrasting observation.

As each test was run from the Naval Research Lab (NRL), Polar is used as a control test location. These particular tests are performed on-site at NRL using the local network infrastructure and accessing the HPC system (Polar) that is onsite. Other tests were performed from the local network infrastructure to other sites via the DREN wide-area network. The file sizes were chosen to get a variety of results from each system. a 50Mb file is small enough to provide an estimated "fastest throughput" measurement, while a 3Gb file is large enough to achieve a long-term throughput measurement.

Each file was transfered to a single site a total of three times per transfer protocol. For example a 50Mb file was transferred to an HPCMP supercomputer three times for FTP using the FileZilla client, three times for SFTP using the FileZilla client, and three times for SFTP using the PuTTY client. The utility server for each HPCMP site was also tested, although these servers do not allow normal FTP. This was done for each file size to get an average transfer time. Each protocol was tested on six HPCMP locations across

Table 2: HPCMP distances in miles

| ARL Excalibur | MHPCC Riptide | NAVO Haise | AFRL Spirit | ERDC Garnet | NRL Polar |
|---|---|---|---|---|---|
| 62 miles | 4,781 miles | 926 miles | 329 miles | 899 miles | 0 miles |

the country to get a variety of geographical distances the file has to travel in order to complete its route (see table:HPCMP distances). This procedure was done for both the current release kit and the pre-release production kit for a total of 378 individual tests in order to note any improvements.

Based on work in IP mapping for the internet, it can be seen that geographic distances between endpoint nodes is a contributing factor in network latency [6]. The distances noted in Table 2 can be correlated to the latency experienced at each HPCMP location. As mentioned before, site utilization can vary slightly, but typical utilization and latency are within a normal range for these tests. For detailed data on each HPCMP site tested, see Table 1.

### 2.3 Windows 7 Results

Since Windows 7 is one of the most common operating system in use from Microsoft at the time of this writing, this data represents a percentage of the userbase affected by any improvements made in throughput speeds from development. The current release kit is used as a baseline comparison against the pre-release production kit.

Table 3: Windows 7 pre-release production kit percentages of baseline

| | 50MB FTP FileZilla | 3GB FTP FileZilla | 50MB SFTP FileZilla | 3GB SFTP FileZilla | 50MB SFTP PuTTY | 3GB SFTP PuTTY |
|---|---|---|---|---|---|---|
| ARL Excalibur | 116% | 126% | 27% | 28% | 82% | 84% |
| ARL Utility | | | 22% | 19% | 100% | 94% |
| MHPCC Riptide | 71% | 59% | 27% | 35% | 92% | 100% |
| MHPCC Utility | | | 26% | 30% | 88% | 100% |
| NAVO Haise | 111% | 69% | 28% | 35% | 100% | 62% |
| NAVO Utility | | | 35% | 32% | 88% | 79% |
| AFRL Spirit | 73% | 80% | 43% | 39% | 21% | 83% |
| AFRL Utility | | | 41% | 45% | 100% | 111% |
| ERDC Garnet | 65% | 49% | | | 86% | 90% |
| ERDC Utility | | | | | 101% | 90% |
| NRL Polar | 133% | 97% | | | | |

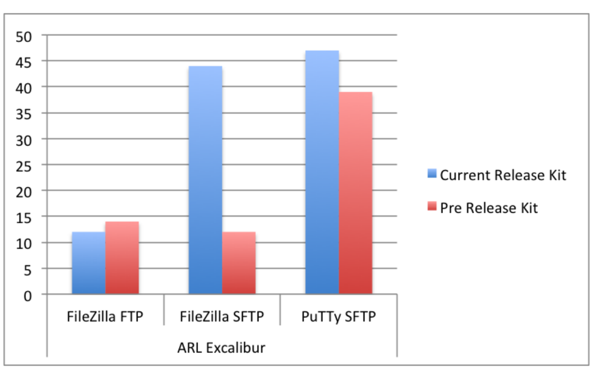
Looking at Table 3, we see that overall, the majority of tests run on the pre-release production kit ran in less time, compared to the same HPCMP site using the current release kit. The baseline is known to be the time to completely transfer a file usign the current release kit. For example, in Table 3 above, row one column three, we note that the 50MB file transfer for ARL's Excalibur using SFTP completed in 27% of the time compared to the ARL Excalibur baseline using the current release kit. That means if the current release kit completed the file transfer in 100 seconds, the pre-release production kit completed the transfer in 27% of that time, which would be 27 seconds. The lower the percentage, the higher increased performance occurred using the pre-release production kit. There are points of interest where the pre-release production kit took longer to complete a transfer than the current release kit. Looking at Table 3 row one, column one, the 50MB file transfer to ARL Excalibur using the pre-release production kit completed in 116% of the time it took the current release kit.

These anomalies happen primarily when the FTP protocol is used (see Figures 2a and 2b) . This is unusual, as FTP transfers do not need the extra overhead that SFTP entails. If this were an operating system rooted problem, the loss of performance using the pre-production release kit would be visible across all HPCMP sites. However, with only a few sites showing loss of performance, it could point to other problems at the time such as network congestion or geogrpahic latency.

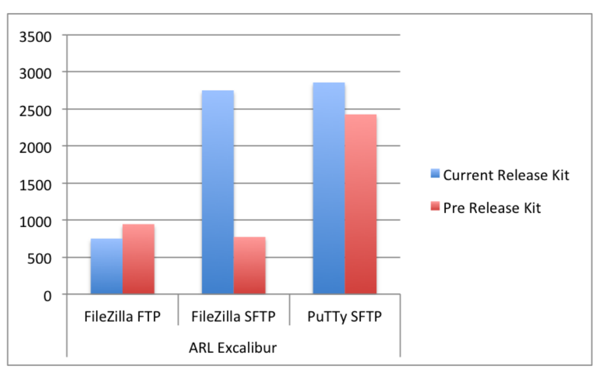
This trend of improvement can be seen in the majority of the tests for Windows 7, although a few points of interest occur where the results show poorer performance. For some of these sites, NRL Polar for example, the results shown in Table 3 may be misleading. For this particular case, a completion time of four seconds on the pre-release production kit, compared to three completion time on the current release kit, will produce the results shown above.This one second time difference jumped the measurement by 33%; but looking at the bigger picture, a difference of one second between the current release and pre-release production kit may not be a significant statistic. It should be noted, this misleading percentage only occurs on the control site at NRL's Polar, or in rare cases on other sites. After looking at the raw data, the small variations in shorter transfer times only apply to the control test. All other HPCMP location tests took enough time to provide statistically clear data.

Fig. 2: ARL Windows 7 Tests (seconds)

(a) 50MB File Transfers




(b) 3GB File Transfers



Apart from the few instances where the pre-release production kit performance is worse than the current release kit, the pre-release production kit increases file transfer performance by a significant amount (see Figure 2). The pre-release kit shows an average of 68% increased performance over the current release kit when using the FileZilla client. A smaller percentage of increased performance occurs over PuTTY's SFTP transfers, but it is still a significant amount.

The trend in increased performance is primarily noted in SFTP transfers using both the FileZilla and PuTTY clients. The improvements from the tests vary from as high as 81% increased performance to 0% increased performance. During only two instances the pre-release kit performance worsens. Although increased performance still occurs, the substantiality of this increase is drastically less when using the PuTTY client rather than the FileZilla client. Even though the data shows increased performance, the scale of time is still larger than would be expected for file transfers of this size. Raw data is available is available in Appendix B.

### 2.4 Windows 8.1 Results

Microsoft Windows 7 was released in the summer of 2009. Since then, Microsoft released the next version of Windows, which is 8.1, in 2012. The same tests have been performed on Windows 8.1 to determine if changes in how the operating system handles networking traffic have been made.

Table 4: Windows 8.1 pre-release production kit percentages of baseline

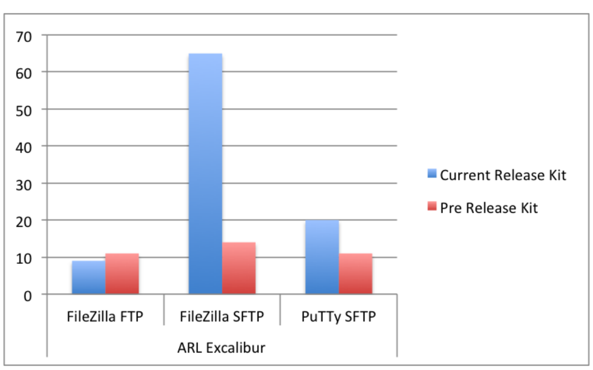
| | 50MB FTP FileZilla | 3GB FTP FileZilla | 50MB SFTP FileZilla | 3GB SFTP FileZilla | 50MB SFTP PuTTY | 3GB SFTP PuTTY |
|---|---|---|---|---|---|---|
| ARL Excalibur | 122% | 113% | 21% | 19% | 55% | 86% |
| ARL Utility | | | 15% | 18% | 85% | 93% |
| MHPCC Riptide | 106% | 98% | 34% | 34% | 115% | 80% |
| MHPCC Utility | | | 33% | 35% | 82% | 90% |
| NAVO Haise | 89% | 86% | 32% | 29% | 80% | 82% |
| NAVO Utility | | | 33% | 83% | 53% | 79% |
| AFRL Spirit | 72% | 80% | 24% | 35% | 87% | 93% |
| AFRL Utility | | | 108% | 100% | 80% | 85% |
| ERDC Garnet | 139% | 87% | | | 92% | 83% |
| ERDC Utility | | | | | 76% | 76% |
| NRL Polar | 133% | 101% | | | | |

Analysis of Table 4 shows that, similar to Windows 7, the majority of improvements occur when using SFTP in FileZilla. The improvment when using the pre-release production kit, on average, comes to about 60%. Again, there are a few points of interest where a loss of performance is present. These anomalies mostly occur when using FTP on the FileZilla client, much like in Windows 7. The poorer performance on NRL Polar is neglible due to the short transfer times to that site. It shows loss of performance even though the difference between current release and pre-release production kits is only one second.

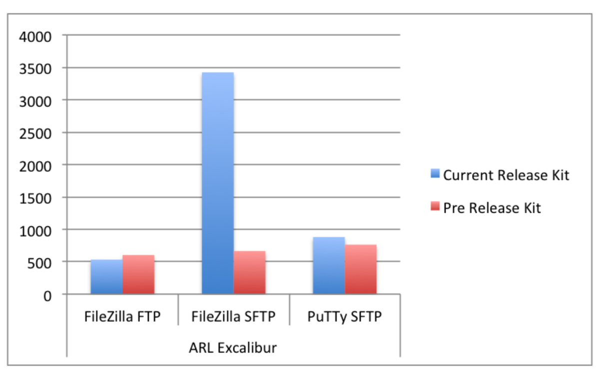
Anomalies of poorer performance is still seen reoccuring on ARL Excalibur, however, the scale of increased performance overall persists on this version of Windows from previous tests (see Figures 3a and 3b). To reiterate, the largest improvement comes when using SFTP on FileZilla. While there is still an increase in performance when using SFTP on PuTTY, it is not as substantial.

Fig. 3: ARL Windows 8.1 Tests (seconds)

(a) 50MB File Transfers




(b) 3GB File Transfers



Performance measurements are relatively similar to Windows 7 even though there was a major version change from Microsoft. This could be due to the underlying network stack configuration not being changed drastically between versions. The only noticable change is that the PuTTY performance, for both the current release and pre-release production kits, is substantially better for Windows 8.1. There was a noticable difference in file transfer speeds when using the pre-release production kit compared to using the current release kit.

### 2.5 Windows 10 Results

With the release of Microsoft Windows 10 in 2015, tests were used to verify results for users who would upgrade their operating systems in the future. These tests will quantify any difference in network traffic handling between the three major Microsoft operating systems.

Based on Table 5, there is a general improvement when using the pre-release production kit over the current release kit. The biggest improvement shows in FileZilla when using SFTP. These improvements range from a 91% increase to a 0% increase when the transfer takes the same amount of time on both kits. This improvement is more consistenly high on Windows 10 than it is for the previous Windows versions.

Table 5: Windows 10 pre-release production kit percentages of baseline

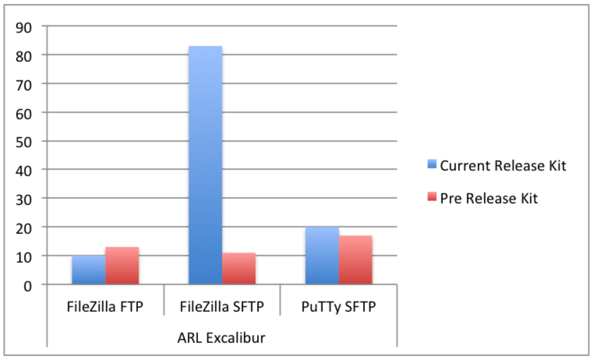
| | 50MB FTP FileZilla | 3GB FTP FileZilla | 50MB SFTP FileZilla | 3GB SFTP FileZilla | 50MB SFTP PuTTY | 3GB SFTP PuTTY |
|---|---|---|---|---|---|---|
| ARL Excalibur | 130% | 110% | 13% | 12% | 85% | 73% |
| ARL Utility | | | 9% | 11% | 91% | 84% |
| MHPCC Riptide | 103% | 111% | 23% | 33% | 99% | 84% |
| MHPCC Utility | | | 30% | 35% | 68% | 71% |
| NAVO Haise | 136% | 98% | 27% | 34% | 87% | 91% |
| NAVO Utility | | | 32% | 36% | 96% | 80% |
| AFRL Spirit | 175% | 84% | 35% | 33% | 85% | 105% |
| AFRL Utility | | | 27% | 20% | 98% | 84% |
| ERDC Garnet | 108% | 95% | | | 45% | 73% |
| ERDC Utility | | | | | 47% | 67% |
| NRL Polar | 166% | 97% | | | | |

As seen in previous versions of Windows, the loss of performance occurs when using FTP methods in FileZilla. This continues the trend established from previous testing. This loss of performance, however, is more drastic than previously seen. The data shows an increase in time differential when a loss in performance occurs.

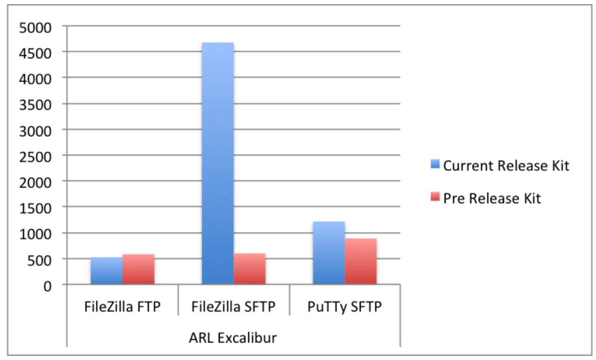
As is the case for Windows 8.1, the PuTTY performance using SFTP is increased compared to Windows 7 (see Figures 4a and 4b). The large increase in performance for SFTP is greater than was seen in previous Windows version, with completion times relatively 25% of the current release kit. This statistic is seen across all HPCMP sites tested across the country.

Fig. 4: ARL Windows 10 Tests (seconds)

(a) 50MB File Transfers




(b) 3GB File Transfers




When testing with Windows 10, roughly the same results can be seen as in the Windows 8.1 tests. Overall there seems to be an increase in speeds when using the pre-release production kits when considering both FileZilla and PuTTY clients. This may not be the case for future updates, as the underlying network handling could change. However, this data shows that the overall procedure for handling network traffic from Microsoft stays the same with only slight modifications being made throughout the three major versions of Windows.

### 2.6 TCP Optimization

The data shown for each operating system was done with an unmodified Windows instance to obtain ”normal” user results. These statistics would be seen from average users who do not modify their machines for optimal network traffic. Measurements have been made in order to understand how much performance can be gained when a user modifies their TCP registry settings. The changes in times were only apparent when using FTP, as shown in Table 6. The optimized Windows 7 installation had a 23% improvement for the large file test, while there was a 40% improvement for the small file test from ERDC as an example. SFTP transfer times were not affected by these changes so they are not shown.

Table 6: Windows 7 TCP Tweaking Comparison (seconds)

| | 50MB FTP | 3GB FTP |
|---|---|---|
| ERDC Garnet UnOptimized | 88 | 4620 |
| ERDC Garnet Optimized | 54 | 3120 |

Windows has a limited number of registries to change to optimize any kind of TCP traffic. TCP Window Auto-Tuning, Congestion Control Provider, and NetDMA (TCPA) is a short list of examples. TCP Window Auto-Tuning allows the operating system to monitor and configure connection window size for best performance over a TCP connection. While most applications will utilize and benefit from this, in some

cases it may hinder performance from applications that do not support this functionality. This flag should be set to normal. Congestion Control Provider should be set to Compound TCP (CTCP) to be able to adjust the sender's window, in order to optimize bandwidth between sender and receiver. NetDMA allows for the network card to interface with direct memory, so as to save on CPU resources.

When all of these are set, a slight performance increase was found over the normal windows configuration. It allowed for faster throughput on a Kbps scale. While this helps on an operating system level, it does not match the scale of improvements of updated software on an application level.

## 3. CROSS-PLATFORM COMPARISON

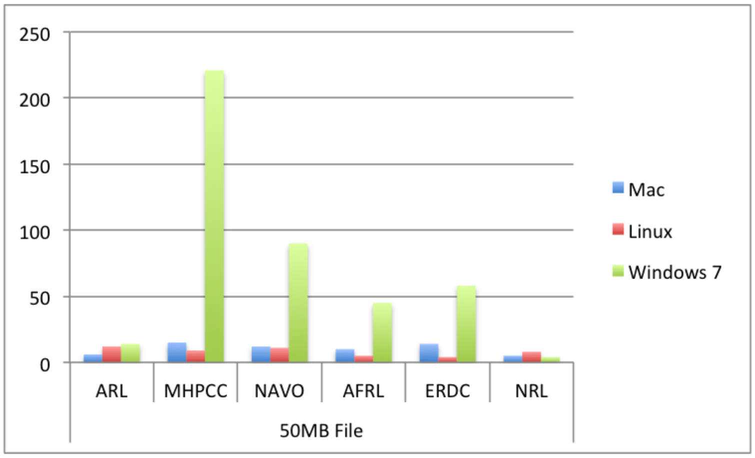
Apart from the variances between Windows versions, other platforms were tested under the same circumstances to give a full picture of differentiating transfer speeds to each site. Both Mac (El Capitan) and Linux (Ubuntu 16.04) showed increased throughput speeds when compared to the Windows results. These tests were executed in the same manner.

As shown in Figure 5, the file transfer times for Windows are substantially longer compared to both Mac and Linux. The transfer times for Windows grow as the specified HPCMP test site distance to NRL increases, while the Mac and Linux results show consistency regardless of distance.

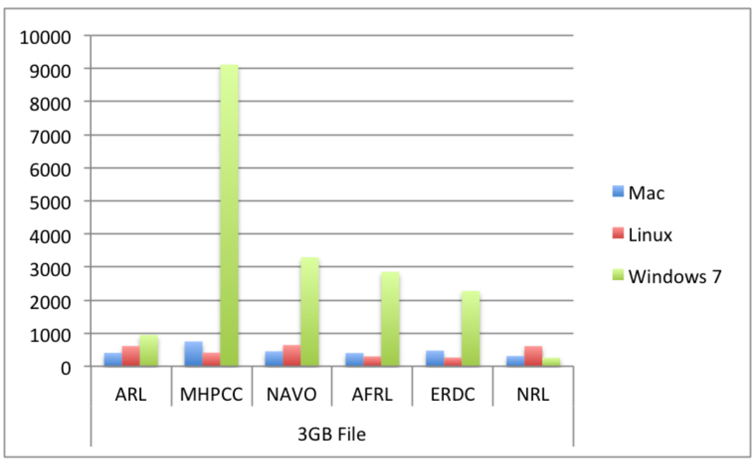
These tests only include command line transfer applications, as Linux and Mac do not have FileZilla included as a file transfer client. The raw data, available in appendix C, shows that the lack of throughput speeds is result of the Windows implementation of network services support. This is evident, as non-Windows operating systems transfer times are much lower.

Fig. 5: All locations OS Comparison (seconds)

(a) 50MB File Transfers




(b) 3GB File Transfers




## 4. CONCLUSION

The data shows there is a correlation between distance to the HPCMP site and throughput speeds achieved. Sites farther away (i.e., MHPCC Riptide) take much longer than closer sites. This is backed up by the network round-trip data listed with each site description. This is due to the fact that the connections are TCP. This means that for every transmission packet window, there has to be a response from the server as an acknowledgment. The larger the distance, the longer it takes to get a response due to travel time.

This difference between utilities, FileZilla and PuTTY, are slight but still noticeable. PuTTY file transfers with SFTP take longer than FileZilla, but is still an improvement from the current release production kits being used. FileZilla using SFTP offers the biggest improvement in throughput speeds by far. It, on average, offers a 60-70% increase in throughput, while PuTTY offers 30% and lower increases. For FTP, the varying results make it hard to define a clear trend. It is seen in figures for each section that in the majority of cases, the pre-release production kit results in lower performance. For all HPCMP sites, SFTP should be prioritized when using the pre-release production client as it offers faster file transfer times.

Based on the results across all systems, OSX and Linux operating systems performed significantly better than the Windows counterpart. This indicates that any users that require high throughput to HPCMP systems should utilize OSX or Linux as a work platform. If those systems are not an option, TCP optimizations should be made on the Windows system of choice for best throughput performance.

Overall, the Kerberos pre-release production kit (v0.66) offers a general improvement in file transfer speeds via various utilities and protocols. This is shown across the board against all HPCMP sites in their respective locations across the country. This also takes in account the geographical distance between client and server. The results across the Microsoft Windows versions are generally comparable. In all cases except Windows 7, PuTTY offers better performance when using the current release kit. FileZilla with SFTP offers the best performance while using the pre-release production kit. Overall, the Kerberos pre-release production kit internally developed and maintained by the HPCMP development team is an improvement over the current release kit.

**REFERENCES**

1. Information Systems and Technology Massachusetts Institue of Technology, "Kerberos: The Network Authentication Protocol" (2017), URL `https://web.mit.edu/kerberos/`.

2. HPCMP, "HPCMP: High Performance Computing Modernization Program" (2013), URL `https://www.hpc.mil/index.php/2013-08-29-16-01-36/about`.

3. FileZilla, "FileZilla: The Free FTP Solution" (2017), URL `https://filezilla-project.org/`.

4. PuTTY, "PuTTY: A Free SSH and Telnet Client" (2017), URL `http://www.chiark.greenend.org.uk/~sgtatham/putty/`.

5. Information Systems and Technology Massachusetts Institue of Technology, "Kftp: Kerberized file transfer protocol (ftp) at mit" (2017), URL `https://ist.mit.edu/software/ftp`.

6. V. N. Padmanabhan and L. Subramanian, "An Investigation of Geographic Mapping Techniques for Internet Hosts," *SIGCOMM Comput. Commun. Rev.* **31**(4), 173–185 (Aug. 2001), ISSN 0146-4833, doi: 10.1145/964723.383073, URL `http://doi.acm.org/10.1145/964723.383073`.

## Appendix A

## WINDOWS 7 RAW DATA

Table A1: Windows 7 current release kit raw data (seconds)

| | FileZilla FTP | | | | FileZilla SFTP | | | | PuTTY SFTP | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 50MB | 3GB | High | Low | 50MB | 3GB | High | Low | 50MB | 3GB |
| ARL Excalibur | 12 | 750 | 4.8 MB/s | 3.5 MB/s | 44 | 2752 | 1.0 MB/s | 890 KB/s | 47 | 2858 |
| ARL Utility | | | | | 54 | 3312 | 1.0 MB/s | 695 KB/s | 39 | 2545 |
| MHPCC Riptide | 308 | 15240 | 256 KB/s | 125 KB/s | 546 | 34200 | 94 KB/s | 70 KB/s | 941 | 25200 |
| MHPCC Utility | | | | | 553 | 33900 | 85 KB/s | 68 KB/s | 954 | 25200 |
| NAVO Haise | 81 | 4920 | 707 KB/s | 570 KB/s | 158 | 9720 | 326 KB/s | 287 KB/s | 271 | 25200 |
| NAVO Utility | | | | | 168 | 10140 | 1.2 MB/s | 270 KB/s | 261 | 17520 |
| AFRL Spirit | 61 | 3568 | 820 KB/s | 463 KB/s | 136 | 8280 | 480 KB/s | 321 KB/s | 920 | 14280 |
| AFRL Utility | | | | | 140 | 8460 | 358 KB/s | 190 KB/s | 226 | 14400 |
| ERDC Garnet | 88 | 4620 | 702 KB/s | 471 KB/s | 168 | 10560 | 290 KB/s | 190 KB/s | 279 | 16500 |
| ERDC Utility | | | | | 168 | 10800 | 420 KB/s | 281 KB/s | 282 | 16380 |
| NRL Polar | 3 | 266 | 13 MB/s | 10.5 MB/s | | | | | | |

Table A2: Windows 7 pre-release kit raw data (seconds)

| | FileZilla FTP | | | | FileZilla SFTP | | | | PuTTY SFTP | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 50MB | 3GB | High | Low | 50MB | 3GB | High | Low | 50MB | 3GB |
| ARL Excalibur | 14 | 945 | 13.2 MB/s | 2.3 MB/s | 12 | 772 | 10.3 MB/s | 3.4 MB/s | 39 | 2427 |
| ARL Utility | | | | | 12 | 630 | 8.5 MB/s | 2.2 MB/s | 39 | 2398 |
| MHPCC Riptide | 221 | 9120 | 848 KB/s | 192 KB/s | 152 | 12120 | 2.1 MB/s | 270 KB/s | 871 | 25200 |
| MHPCC Utility | | | | | 148 | 10380 | 1.7 MB/s | 370 KB/s | 841 | 25200 |
| NAVO Haise | 90 | 3294 | 1.1 MB/s | 532 KB/s | 45 | 3486 | 2.2 MB/s | 778 KB/s | 271 | 15840 |
| NAVO Utility | | | | | 59 | 3316 | 1.4 MB/s | 895 KB/s | 232 | 13920 |
| AFRL Spirit | 45 | 2856 | 1.5 MB/s | 960 KB/s | 59 | 3251 | 1.2 MB/s | 720 KB/s | 201 | 11940 |
| AFRL Utility | | | | | 58 | 3840 | 1.4 MB/s | 820 KB/s | 228 | 16020 |
| ERDC Garnet | 58 | 2278 | 748 KB/s | 532 KB/s | | | | | 242 | 14880 |
| ERDC Utility | | | | | | | | | 285 | 14820 |
| NRL Polar | 4 | 260 | 12.8 MB/s | 10.9 MB/s | | | | | | |

## Appendix B

## WINDOWS 8.1 RAW DATA

Table B1: Windows 8.1 current release kit raw data

| | FileZilla FTP | | | | FileZilla SFTP | | | | PuTTY SFTP | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 50MB | 3GB | High | Low | 50MB | 3GB | High | Low | 50MB | 3GB |
| ARL Excalibur | 9 | 533 | 6.6 MB/s | 1.0 MB/s | 65 | 4523 | 862 KB/s | 800 KB/s | 20 | 881 |
| ARL Utility | | | | | 59 | 3363 | 910 KB/s | 710 KB/s | 14 | 803 |
| MHPCC Riptide | 152 | 9840 | 540 KB/s | 230 KB/s | 540 | 35220 | 90 KB/s | 83 KB/s | 211 | 14220 |
| MHPCC Utility | | | | | 554 | 34200 | 87 KB/s | 83 KB/s | 214 | 14640 |
| NAVO Haise | 55 | 3660 | 1.1 MB/s | 530 KB/s | 168 | 10440 | 305 KB/s | 290 KB/s | 65 | 4500 |
| NAVO Utility | | | | | 164 | 10020 | 290 KB/s | 150 KB/s | 86 | 4320 |
| AFRL Spirit | 44 | 3840 | 1.1 MB/s | 750 KB/s | 144 | 8400 | 339 KB/s | 250 KB/s | 63 | 3840 |
| AFRL Utility | | | | | 143 | 8760 | 330 KB/s | 250 KB/s | 57 | 3720 |
| ERDC Garnet | 46 | 4440 | 1.0 MB/s | 590 KB/s | 191 | 12420 | 280 KB/s | 140 KB/s | 78 | 4380 |
| ERDC Utility | | | | | | | | | 73 | 4620 |
| NRL Polar | 3 | 268 | 13.1 MB/s | 10.3 MB/s | | | | | | |

Table B2: Windows 8.1 pre-release kit raw data

| | FileZilla FTP | | | | FileZilla SFTP | | | | PuTTY SFTP | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 50MB | 3GB | High | Low | 50MB | 3GB | High | Low | 50MB | 3GB |
| ARL Excalibur | 11 | 604 | 5.9 MB/s | 4.2 MB/s | 14 | 665 | 6.5 MB/s | 4.5 MB/s | 11 | 764 |
| ARL Utility | | | | | 9 | 628 | 7.2 MB/s | 4.3 MB/s | 12 | 754 |
| MHPCC Riptide | 162 | 9660 | 500 KB/s | 240 KB/s | 188 | 12120 | 820 KB/s | 185 KB/s | 244 | 11400 |
| MHPCC Utility | | | | | 188 | 12120 | 385 KB/s | 220 KB/s | 177 | 13320 |
| NAVO Haise | 49 | 3177 | 1.4 MB/s | 850 KB/s | 55 | 3096 | 1.5 MB/s | 600 KB/s | 52 | 3720 |
| NAVO Utility | | | | | 55 | 8404 | 1.3 MB/s | 890 KB/s | 54 | 3434 |
| AFRL Spirit | 32 | 3078 | 1.2 MB/s | 880 KB/s | 35 | 3004 | 1.3 MB/s | 900 KB/s | 55 | 3600 |
| AFRL Utility | | | | | 155 | 8760 | 320 KB/s | 250 KB/s | 46 | 3179 |
| ERDC Garnet | 64 | 3900 | 1.3 MB/s | 550 KB/s | | | | | 72 | 3660 |
| ERDC Utility | | | | | | | | | 56 | 3543 |
| NRL Polar | 4 | 273 | 21 MB/s | 10 MB/s | | | | | | |

# Appendix C

## WINDOWS 10 RAW DATA

Table C1: Windows 10 current release kit raw data

| | FileZilla FTP | | | | FileZilla SFTP | | | | PuTTY SFTP | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 50MB | 3GB | High | Low | 50MB | 3GB | High | Low | 50MB | 3GB |
| ARL Excalibur | 10 | 525 | 7.1 MB/s | 3.0 MB/s | 83 | 4680 | 670 KB/s | 476 KB/s | 20 | 1216 |
| ARL Utility | | | | | 105 | 4920 | 620 KB/s | 310 KB/s | 12 | 703 |
| MHPCC Riptide | 153 | 10800 | 380 KB/s | 170 KB/s | 539 | 34500 | 80 KB/s | 64 KB/s | 207 | 13740 |
| MHPCC Utility | | | | | 544 | 34800 | 85 KB/s | 70 KB/s | 261 | 17580 |
| NAVO Haise | 47 | 3579 | 1.1 MB/s | 850 KB/s | 172 | 10260 | 275 KB/s | 120 KB/s | 70 | 4380 |
| NAVO Utility | | | | | 173 | 10500 | 250 KB/s | 100 KB/s | 65 | 4560 |
| AFRL Spirit | 36 | 3600 | 1.3 MB/s | 575 KB/s | 145 | 10080 | 390 KB/s | 310 KB/s | 77 | 3478 |
| AFRL Utility | | | | | 146 | 13920 | 330 KB/s | 190 KB/s | 50 | 4080 |
| ERDC Garnet | 62 | 3780 | 1.2 MB/s | 720 MB/s | 62 | 11040 | 270 KB/s | 230 KB/s | 162 | 5280 |
| ERDC Utility | | | | | 183 | 11580 | 290 KB/s | 220 KB/s | 119 | 5040 |
| NRL Polar | 3 | 271 | 11.2 MB/s | 7.0 MB/s | | | | | | |

Table C2: Windows 10 pre-release kit raw data

| | FileZilla FTP | | | | FileZilla SFTP | | | | PuTTY SFTP | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 50MB | 3GB | High | Low | 50MB | 3GB | High | Low | 50MB | 3GB |
| ARL Excalibur | 13 | 582 | 6.4 MB/s | 3.5 MB/s | 11 | 600 | 8.0 MB/s | 4.5 MB/s | 17 | 889 |
| ARL Utility | | | | | 10 | 544 | 10.8 MB/s | 5.0 MB/s | 11 | 595 |
| MHPCC Riptide | 158 | 12060 | 408 KB/s | 320 KB/s | 129 | 11400 | 800 KB/s | 280 KB/s | 205 | 11640 |
| MHPCC Utility | | | | | 166 | 12480 | 400 KB/s | 250 KB/s | 180 | 12540 |
| NAVO Haise | 64 | 3532 | 1.3 MB/s | 780 KB/s | 37 | 3539 | 1.2 MB/s | 730 KB/s | 61 | 4020 |
| NAVO Utility | | | | | 57 | 3840 | 1.2 MB/s | 500 KB/s | 63 | 3660 |
| AFRL Spirit | 63 | 3048 | 870 KB/s | 450 KB/s | 50 | 3330 | 1.2 MB/s | 540 KB/s | 66 | 3660 |
| AFRL Utility | | | | | 40 | 2846 | 1.7 MB/s | 620 KB/s | 49 | 3464 |
| ERDC Garnet | 67 | 3600 | 1.2 MB/s | 660 KB/s | | | | | 73 | 3900 |
| ERDC Utility | | | | | | | | | 56 | 3379 |
| NRL Polar | 5 | 265 | 15 MB/s | 10 MB/s | | | | | | |

## Appendix D

## MAC AND LINUX RAW DATA

Table D1: Mac and Linux Raw Data (seconds)

| | Mac Transfers | | Linux Transfers | |
|---|---|---|---|---|
| | 50MB | 3GB | 50MB | 3GB |
| ARL Excalibur | 6 | 405 | 12 | 612 |
| ARL Utility | 6 | 403 | 8 | 424 |
| MHPCC Riptide | 15 | 750 | 9 | 412 |
| MHPCC Utility | 23 | 1702 | 8 | 523 |
| NAVO Haise | 12 | 453 | 11 | 642 |
| NAVO Utility | 13 | 425 | 7 | 371 |
| AFRL Spirit | 10 | 401 | 5 | 299 |
| AFRL Utility | 12 | 502 | 4 | 264 |
| ERDC Garnet | 14 | 474 | 4 | 265 |
| ERDC Utility | 12 | 509 | 5 | 477 |
| NRL Polar | 5 | 311 | 8 | 610 |